Onboard Adaptive Learning for Planetary Surface Rover Control in Rough Terrain*

Terry Huntsberger, Hrand Aghazarian, and Edward Tunstel

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA, 91009
{Terry.Huntsberger, Hrand.Aghazarian, Edward.Tunstel}@jpl.nasa.gov

***Abstract* – Current and future NASA robotic missions to planetary surfaces are tending toward longer duration and are becoming more ambitious for rough terrain access. For a higher level of autonomy in such missions, the rovers will require behavior that must also adapt to declining rover health and unknown environmental conditions. The MER (Mars Exploration Rovers) called Spirit and Opportunity have both passed 200 days of life on the Martian surface, with possible extensions to 300 days and beyond depending on rover health. Changes in navigational planning due to degradation of the drive motors as they reach their lifetime are currently done on Earth for the Spirit rover. The upcoming 2009 MSL (Mars Science Laboratory) and 2013 AFL (Astrobiology Field Laboratory) missions are planned to last 300-500 days, and will possibly involve traverses on the order of multiple kilometers over challenging terrain. This paper presents an adaptive control algorithm for onboard learning of weights within a free flow hierarchy (FFH) behavior framework for autonomous control of planetary surface rovers that explicitly addresses the issues of rover health and rough terrain access. We also present the results of some laboratory and field studies.**

Index Terms – Adaptive behavior, onboard learning, planetary surface rovers

I. Introduction

High-value science data acquisition on rough terrain (example shown in Figure 1(a)) is beyond the capabilities of current NASA rover designs. Although the JPL technology prototype rover SRR (Sample Return Rover) shown in Figure 1(b) has the ability to mechanically adapt itself to changing terrain by varying its shoulder angles [1, 2, 3, 4], such an operation requires a high level of adaptability in the onboard control algorithms during the mission in order to maintain the health of the rover. In addition, as the mission progresses, the onboard control must also adapt to degraded performance due to wear-and-tear on components such as the steering and drive mechanisms.

We previously developed a behavior-based framework called BISMARC (Biologically Inspired System for Map-based Autonomous Rover Control) to address these concerns at the system level by treating rover motion, rover health, and resource management within a FFH (free flow hierarchy) [5, 6, 7]. BISMARC has demonstrated robust performance for a number of different simulated mission scenarios including multiple cache retrieval [8], fault tolerance for long duration missions [9], and site preparation [10].

The major limitation in the original implementation of BISMARC was the use of fixed weights in the FFH, which effectively made it unable to adapt to situations outside of the original world model. This paper presents an onboard mechanism for learning weights within the FFH that will adapt not only to the dynamic environment around the rover, but also to the degradation of mechanical components during the mission lifetime. Our goal in this research is not to find the optimal weight adaptation policy, but instead one that is "good enough" to maintain rover health while still achieving high level goals.

(a) (b)

Figure 1. Planetary surface terrain and technology example for autonomous access to high risk, scientifically interesting regions. (a) Mars cliff-face with signs of water outflows; (b) JPL technology prototype of a terrain-adaptive reconfigurable rover.

The next section discusses some background and related work from ethological studies of animal behavior and learning mechanisms. This is followed by a brief description of the organization of the action selection mechanism of BISMARC, followed by a discussion of the learning mechanism of the system. We close with experimental studies and conclusions.

II. Background & Related Work

Ethologists analyze animal behavior and develop models and explanations based upon external observations. The conceptual models and ideas that have emerged from the past half-century of ethology are quite useful as foundations for realizing intelligent behavior synthesis in robots [6, 11]. Several concepts that are thought to contribute to animal intelligence and adaptability include hierarchical organizations of behavior, concurrent activation and coordination of motivational tendencies (e.g., multi-behavior action selection), and individual behavior excitation and inhibition via thresholds.

The hierarchical nature of behavior is supported by a host of similar conceptual models of motivational control of behavior in animals. Examples include Tinbergen's hierarchy of instinct centers [12], Baerends' hierarchical

* The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. An earlier version of the manuscript appeared in [7].

decomposition of herring gull incubation behavior [13], and MacLean's triune brain concept [14] among others. These models postulate a decomposition of behavior into high and low levels of instinct/motivation in which higher levels modulate lower-level activities to achieve a variety of behavioral expressions. Brooks used this type of hierarchy in the design of his subsumption control architecture [15].

Prior research by Tyrrell [16] and Bryson [17] demonstrated superior performance of a hierarchical system for action selection over purely reactive systems. In particular, the agents in the Edmund system of Bryson [17] are built as related sensing and action functions that exhibit selective attention with the payoff of a higher efficiency than the modified Rosenblatt and Payton (RP) mechanisms of Tyrrell [16, 18]. A comprehensive overview of action selection systems can be found in Bryson [19]. Although BISMARC uses the modified RP mechanisms, the nodes in the FFH perform operations that are more sophisticated than simple combination. In some sense, they are closer to the *competence* structures of Bryson [17], in that a collection of plan elements are organized as a prioritized finite state machine whose outputs converge on a specific goal. These nodes have undergone extensive evaluation at the modular level either through field or mission testing.

To date, there has been very little research into learning for hierarchical action selection systems that are typically characterized by multiple, possibly conflicting goals [20]. The dominant learning strategy for single goal achievement such as robotic navigation has been reinforcement learning (RL), an unsupervised method that seeks to maximize a reward signal based on the utility of pairings of input and output states and their subsequent actions [21, 22, 23]. One of the most popular RL algorithms is Q-learning [24] and its variations such as Q-PSP [25], and hierarchical Q-learning [26]. RL algorithms typically suffer from slow convergence, large state spaces, and difficulties in handling uncertain sensory inputs.

Continuous valued versions of the Q-learning algorithm have been developed to address the large state space problem [27, 28, 29]. These works used a continuous Q-value derived from neural networks or other function approximation methods. The state space concerns were also addressed for deterministic environments using a forgetting mechanism in a penalty-based hierarchical Q-learning algorithm, which reduces the amount of state information that an agent must maintain by using a low level agent to maintain local state information and a high level agent to maintain global state [30, 31]. Most of the RL studies to date have been confined to simulations and interior navigation in 2-D environments.

Most recently, learning of sequential behaviors for goal satisfaction through a blend of static and dynamic behavioral motivation modules has been demonstrated in simulation and on a commercially available AmigoBot in a lab setting [32, 33]. This analysis used state prediction following an action to learn the sequential behaviours. However, in the case of planetary surface rover operations, the relationship between an action and a subsequent state is difficult to derive since it is closer to a non-deterministic process due to interactions with the terrain. However, of particular importance in the behavioral sequence study [32, 33] was the use of short-term memory (STM) and long-term memory (LTM) to store successful behavioral sequences during learning. Memory encoding is an effective technique for limiting the time needed for on-line learning, and is used in the BISMARC learning algorithm.

An alternate learning system to Q-learning and its variants that performs in the presence of a multiple conflicting goals where subtasks are only partially satisfied is W-learning [34]. W-learning is based on compromise or negotiated decision making between agents , and is a memory efficient method that adapts weighted activation levels for action selection [34]. As such, it is more suited for operation onboard planetary surface rovers than traditional or hierarchical Q-learning systems, and a temporally prioritized modification of it is running under BISMARC.

In ethological terms, activation levels or weights can be thought of as conveyors of motivational tendencies for individual behaviors. They serve as a form of motivational adaptation since they cause a control policy to dynamically change in response to goals, sensory inputs, and internal state. Such behavioral flexibility permits adaptation to new environments and degradations in performance over time. We exploit this flexibility in an onboard mechanism for learning weights [7] that will adapt not only to the dynamic environment around a rover, but also to degradations of mechanical component performance (e.g., rover wheel motors) during long duration missions. The next section reviews BISMARC and introduces the FFH for the rough terrain access mission scenario.

III. BISMARC ORGANIZATION

An example of the action selection mechanism used in BISMARC is shown in Figure 2 for a rough terrain navigation mission that is used for the experimental studies reported in this paper. The rectangular boxes represent behaviors and the ovals are sensory inputs (either fixed, direct, or derived). At the top are the high level behaviours: *Don't Tip Over, Go to Goal, Avoid Obstacles*, *Preserve Motors, Warm Up*, *Get Power,* and *Sleep at Night*. These goals are related to both task and rover safety. For example, since most planetary surface rovers have only visual sensors for navigation, the sensory input for *Proximity to Night* is derived from knowledge of the sun's position and forces the rover to sleep at night by weighting the input to *Sleep at Night* heavier (4.0) than any other behavior in the hierarchy. The *Avoid Obstacles* behavior uses the output of an onboard local navigation algorithm as recommendations for viable paths. The rovers are equipped with solar panels and the *Rest* behavior allows the batteries to recharge if the sun is up. The *Rest* behavior is also used to cool down the motors for *Preserve Motors* if they are working too hard going up a steep slope, or to stop and turn on the heaters for *Warm Up* if the internal temperature of the rover drops below a safety threshold.

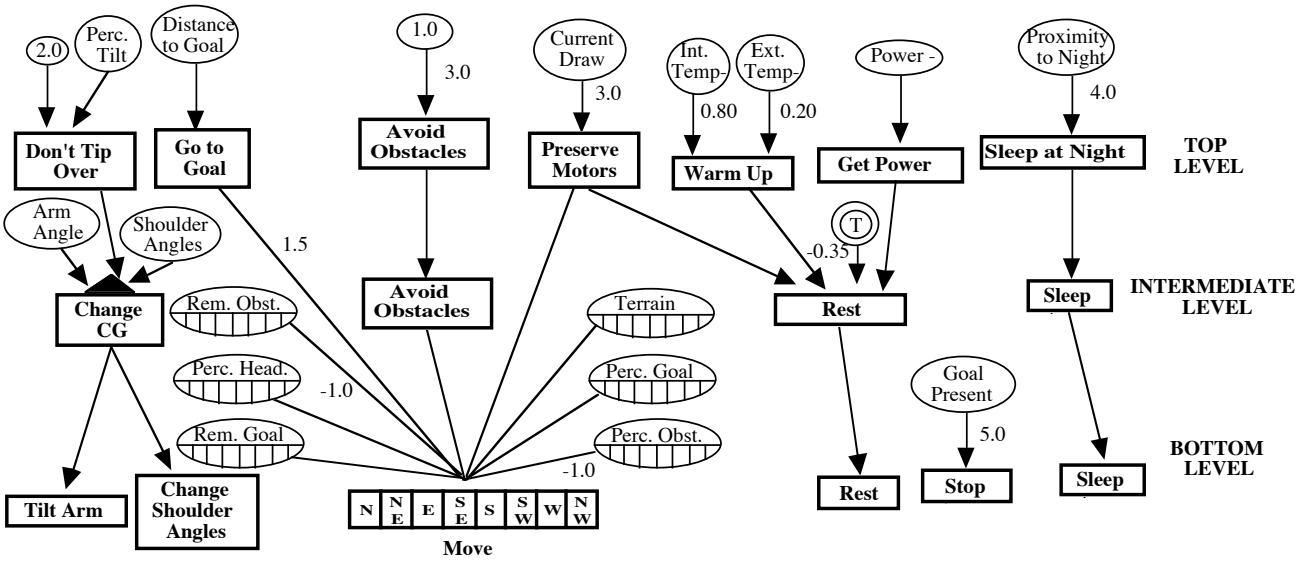

Figure 2: Free-flow hierarchy action selection mechanism for rough terrain navigation mission scenario. Ovals represent inputs derived from sensory stimuli, rectangular boxes are behaviors, and double ovals are temporal penalties. All weights on inputs to behaviors are 1.0 unless otherwise noted. Segmented boxes and ovals represent directional inputs (only cardinal directions shown but in practice continuous coverage). See text for further details.

The intermediate level *Change CG* behavior is an example of a sophisticated combination behavior mentioned in Section 1 that works to shift the center of gravity of the rover (see Figure 1(b)) much like an animal does while climbing on a steep slope. This behavior is implemented using a finite state machine based on a well-tested algorithm for pose reconfiguration [1, 2, 3, 4] The algorithm uses the onboard gyroscopes and accelerometers, which would be equivalent to the inner ear mechanism in mammals for roll and pitch determination. Recommendations for shoulder angle and arm end position changes to help stabilize the rover are generated and passed on to the bottom level behaviors.

The intermediate level behaviors are designed to interact with both the STM, which corresponds to perceived sensory stimuli, and the LTM, which encodes remembered sensory information. Control loops are prevented through temporal penalties (shown as T-ovals in Figure 2) that constrain the system to only repeat a behavior a predetermined number of times. The bottom level behaviors in the hierarchy fuse the sensory inputs and the weighted activations of the higher level behaviors in order to select appropriate actions for rover safety and goal achievement. The rover will continue to move until it achieves the goal position as determined by a rover localization algorithm [35] shown as the *Goal Present* input to *Stop* in Figure 2, or its health deteriorates due to dead batteries, freezing, burned out motors, or tipping over.

BISMARC's map-based LTM (Long Term Memory) is similar to hippocampus place cells. Landmarks corresponding to obstacles and goals are extensively mapped and stored for comparison to perceived inputs, with a probabilistic update of memories based on the positional variance of the rover and the match strength of the current perception to memory contents. A LTM landmark is encoded as a four-byte field that includes relative height of the landmark (2 bytes), actions leading to the landmark (1 byte), and accelerometer readings on the robot (1 byte). A similar approach is the coupled goal/representation framework of [36, 37]. Another alternate approach is an occupancy grid that gives dense coverage of the environment, but doesn't scale well for long duration planetary surface missions [38].

IV. Learning Mechanism

Learning mechanisms for planetary surface rovers have the same requirements as terrestrial robots [39]: (1) noise immunity, (2) fast convergence, (3) incrementality (improving performance while learning), (4) tractability (iterations of algorithm doable in real-time), and (5) groundedness (information limited to onboard sensors). In particular, the fast convergence and tractability requirements are key for planetary surface rovers because they are typically computationally challenged (i.e., MER uses a 27Mhz CPU) due to power constraints. We address (2) and (4) through a behavior decomposition process similar to the use of *heterogeneous reward functions* developed by [40]. We give the details of the reward function for updating the weights for the *Move* behavior (see Figure 2) in this section. For point (3) we use the W-learning algorithm of Humphrys [34] supplemented with a dynamic reward function directly related to rover health. For (1) we use a sequence memory similar to that of McCallum [41, 42] and Michaud and Mataric [43] Finally, we restrict our inputs to onboard sensors only, as stipulated in point (5).

The weights on the links between modules are usually heuristically set based on mission goals. These goals are specified at a relatively high level without complete knowledge of the operating environment of the rover. There is however a priority derived from mission risk mitigation requirements explicitly included in the relative size of the weights. The maximum activation of the high level behaviors are weighted to give the highest priorities to rover preservation. In order of highest priority to lowest these are *Sleep at Night*, *Avoid Obstacles*, *Preserve Motors*, *Don't Tip Over*, *Get Power*, and *Warm Up*. In addition, rover health will degrade as the mission progresses, and weights

chosen at full health may no longer be appropriate. Rover health is defined in Equation (1) as:

$$\text{rover\_health} = \left[w_p \,\text{power} + w_{mc}(1 - \text{motor\_current})\right]\left[\frac{(\text{AGE\_MAX} - \text{age})}{\text{AGE\_MAX}}\right] \qquad (1)$$
$$w_p + w_{mc} = 1$$

where *power* is the current battery levels, *motor_current* is the current draw on the motors, *AGE_MAX* is the maximum expected lifetime for the rover, *age* is the current age of the rover, and w_p and w_{mc} are weights (currently both set to 0.5 since dead batteries are as lethal as burned-out drive motors). A dynamic reward function is defined in Equation (2) goal achievement for each step:

$$\text{reward} = w_{rh}\Delta\text{rover\_health} + w_{ga}\Delta\text{goal\_achievement} \qquad (2)$$
$$w_{rh} + w_{ga} = 1$$

where Δ is the change, and w_{rh} and w_{ga} are weights (currently set to 0.65 and 0.35 based on the relative importance of health and goal achievement determined experimentally).

Learning is only enabled in the weights on the links feeding into the *Move* behavior at the lowest level in the FFH shown in Figure 2. This is done in order to maintain the rover safety embodied in the relatively high priorities of the *Sleep at Night*, *Get Power*, and *Warm Up* high level behaviors. A modified version of the W-learning algorithm of Humphrys [34] is used in BISMARC to dynamically update the weights. In W-learning, agents suggest their actions with a weight *W* and the maximum weight is chosen as the leader. In our case there are three behaviors vying for control of *Move*: *Go to Goal*, *Avoid Obstacles*, and *Preserve Motors*. W-learning uses the difference between the predicted reward $\mathcal{P}$ and the actual reward $\mathcal{A}$ to determine which weights are to be updated [34].

Humphrys used a genetic algorithm run off-line to determine his reward functions. We instead use the expression in Equation (2) in order to capture the true change in the rover health through an action (the motor currents and battery levels are read in real-time). Rover behavior is extensively studied prior to launch through both laboratory and field trial studies, so the predicted changes in rover battery levels and motor currents are known for rover movement and are actually used for resource management planning during the current MER missions.

A small sampling of the predicted rewards is shown in Figure 3 for typical rover behavior. The reward (1) for movement towards the goal on even terrain from a start position is the highest since rover health has a minimal change compared to progress towards the goal. As progressively steeper slopes are attempted, the rewards (2-3) start out being positive since progress towards the goal is still outweighing the impact on rover health, but become more and more negative (4-5) as the steepness increases. Backing-off the slope has a negative reward (6-7) for the relatively benign slopes since the rover health improvement in rover health is outweighed by the lack of progress towards the goal, becoming positive (8-9) for the steeper slopes. The reward (10) for driving sideways is slightly negative since movement away from the goal outweighs the minimal impact on rover health. The last reward (11), driving away from the goal, is a large negative value primarily due to the movement in a direction totally opposite the goal.

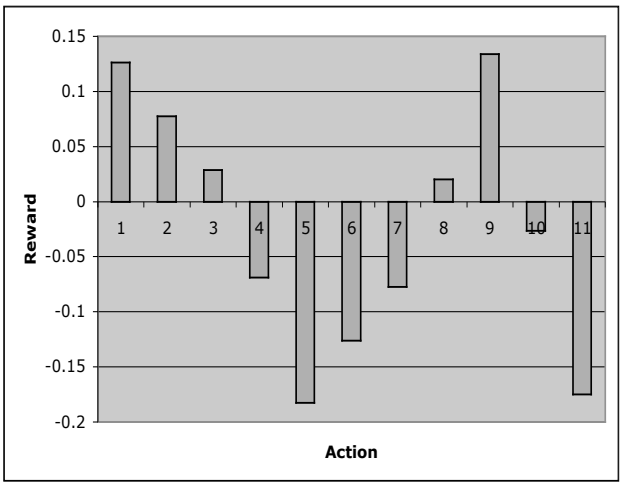

Figure 3: Sample of rewards derived using Equation (2) for a number of different types of actions (in reality the action space is continuous). Reference point is rover health of 1.0 except in cases where the rover is reacting to a current situation (i.e., backing off climbing a slope). All moves are with respect to the goal actions are: (1) normal driving on even terrain, (2) driving up a 5° slope, (3) driving up a 10° slope, (4) driving up a 25° slope, (5) driving up a 45° slope, (6) backing-off a 5° slope drive, (7) backing-off a 10° slope drive, (8) backing-off a 25° slope drive, (9) backing-off a 45° slope drive, (10) driving sideways, and (11) driving backwards. The weights were both set to 0.5 in Equation (1) and to 0.65 and 0.35 respectively for the change in rover health and change in relation to goal in Equation (2).

The *Move/Tilt Arm/Change Shoulder Angles*, *Rest*, *Stop*, and *Sleep* actions at the lowest level in the FFH shown in Figure 2 are mutually exclusive and the action with the maximum activation is chosen using a competitive action selection. The *Tilt Arm, Change Shoulder Angles,* and *Move* actions at the lowest level in the FFH shown in Figure 2 can be done simultaneously, so they are treated as a unit during the action selection process. However, progress towards the goal will be compromised if the rover tips over, so there is a dynamic relationship between the two high level goals of *Go to Goal* and *Don't Tip Over*. The W-learning algorithm is applied to the links feeding into the *Move* behavior in the hierarchy with a time delay between activations. The *Don't Tip Over* behavior activation occurs in the first time slice, followed by the *Go to Goal* weight updates and activation. This maintains the rover health, while at the same time making progress towards the goal. Another instance where this time delay process is applied is in the relative direction that the rover moves. In order to *Preserve the Motors*, the rover will attempt to climb a steep incline, and either back off, go sideways, or rest if the perceived motor currents in the rear wheels are too high. If the weights are not dynamically adjusted, this could lead to dithering where the rover attempts to climb, backs off, and then attempts to climb in the same direction. Adaptive weighting using the W-learning algorithm changes the

direction of attack, since dithering compromises progress towards the goal. For this situation, there is a time delay between application of W-learning to the two incoming links of *Go to Goal* and *Preserve Motors*, with *Preserve Motors* occurring first, followed by *Go to Goal*.

Although our convergence times are typically within 500ms, it is still desirable to limit CPU cycles devoted to learning if it may not be needed. Noise in the sensors can lead to state aliasing where the same sequence of state transitions experienced previously is not recognized. One possible solution to this problem is to provide a memory to the system [41, 42, 43, 44]. We maintain a fixed number of memory traces (currently 100) of limited length (currently 25 steps) of the most recent experiences of the rover. As new experiences come in they are checked for similarity to previous sequences and merged. In the event that the behavior sequence is new, the oldest traces are deleted. These traces are organized using the tree structure developed by Michaud and Mataric [43, 44]. Rather than use these traces to trigger alternate behaviors as done by Michaud and Mataric, we instead use them to seed the W-learning process with the sequence of expected rewards. We have seen a speedup of a factor of two in our step-wise learning.

V. Experimental Studies

In order to determine the utility of BISMARC for planetary surface operations in rough terrain, we have run three different types of experimental studies: (1) 2000 simulated rough terrain navigation missions, (2) 50 laboratory sequences with SRR, and (3) 4 sequences with SRR in natural terrain in the Arroyo Seco outside JPL. We have attempted to match the fidelity of the simulation models for terrain and rovers to those used for the laboratory and field studies.

A. Simulation

The first series of experimental studies used simulated terrain based on MOLA (Mars Orbiter Laser Altimeter) data from the Dao Valis region of Mars, which had slopes of up to 65°. A view of the SRR during one of the simulation runs is shown in Figure 4. Mission success was defined as the attainment of the randomly selected goal position without dying due to freezing, dead batteries, burned out motors, or tipping over. The experimental setup included:

- Random starting and goal positions
- Timestep of 0.1s
- 10% loss of traction in rocky terrain
- 1 sq. km study area (5 cm resolution)
- Top speed of 30 cm/sec

The model of SRR matches the physical platform and has two sets of stereo cameras, one body-mounted and one mast mounted, a 3 DOF (degrees of freedom) manipulator and a twelve week battery lifetime supplemented with solar panels.

Our studies had a 95.9% mission success with the onboard adaptive learning mechanism, and a 43% success rate without the adaptive learning. The primary failure mode (3.8%) for the system with learning enabled was dead batteries, which from a mission standpoint would indicate a need for larger solar panels. An analysis of the 57% of the missions that failed with no learning enabled gives:

- Tipping over - 27%
- Dead batteries - 15%
- Burned out motors - 9%
- Freezing - 6%

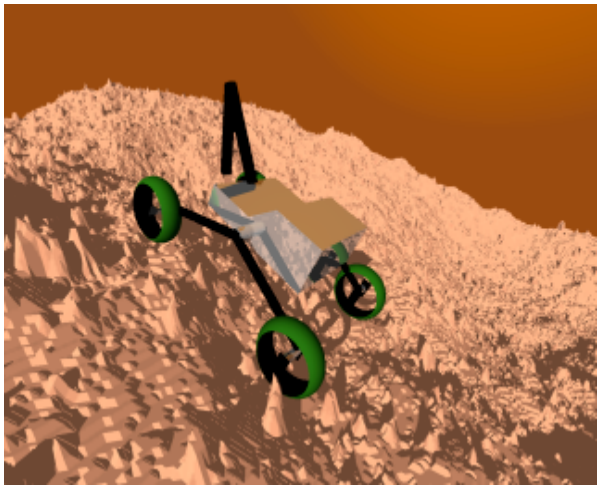

Figure 4. The SRR climbing a 35° slope in simulated terrain derived from MOLA data in the Dao Valis region of Mars. The model of the rover contained full kinematics and dynamics and used a probabilistic slip assumption. The FFH shown in Figure 2 was used for control and adaptive learning for 2000 simulation runs.

Since 27% of the missions failed due to tipping over, the initial weights for inputs to *Move* were set too high, giving an overall bias to the *Get to Goal* behavior over rover safety related behaviors such as *Don't Tip Over*.

B. Laboratory

The second set of experimental studies was run in the Planetary Robotics Lab (PRL) at JPL and used the JPL technology prototype rover SRR shown in Figure 5. SRR has independently articulated shoulders, which allow it to dynamically change its pose and lean much like an animal does on sloped terrain. The full range of shoulder movement is shown in Figure 5. SRR also has independent four-wheel drive and independent four wheel steering enabling it to travel sideways.

One of the experimental runs is shown in Figure 6, where we have set up a worse case scenario of opposing hills and valleys for the rover. The SRR (Sample Return Rover) successfully negotiated the course based on a subnet of the

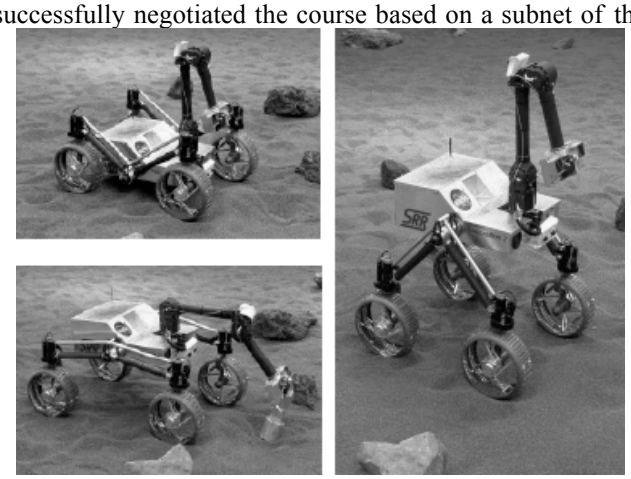

Figure 5. Sample Return Rover (SRR) range of hardware adaptation including clockwise from upper left - the lowest range of the shoulder articulation, the highest range of shoulder articulation, and the mid-range of shoulder articulation coupled with extended arm movement.

full hierarchy shown in Figure 2. This subnet included the *Don't Tip Over, Go to Goal, Avoid Obstacles*, and *Preserve Motors* top level nodes. The *Warm Up*, *Get Power,* and *Sleep at Night* top level node activation levels were all set to zero since the interior of the lab was warm and not exposed to the sun.

Another series of laboratory trials used a ramp set at a 65° slope with the rover positioned at the bottom. The goal position was on the other side of the ramp, which was beyond SRR's stability capabilities to climb even with shoulder reconfiguration. Initially the rover attempted to climb the slope, but repeatedly backed off and then tried again. This behavior can be traced to the combination of *Go to Goal, Avoid Obstacles,* and *Preserve Motors* using the default weights. The learning algorithm progressively reduced the *Go to Goal* weight from 1.5 to 0.45 while at the same time increasing the weights of *Go to Goal* and *Avoid Obstacles* from 1.0 to a high-water mark of 1.6, which caused the rover to try to skirt the ramp by moving sideways while still maintaining movement towards the goal. Although adaptation of the *Avoid Obstacles* weights lagged behind those of the *Preserve Motors*, the ramp was eventually seen as an obstacle and the obstacle avoidance behavior kicked in. As the rover cleared the side of the ramp it then started movement towards the goal due to the *Go to Goal* behavior output dominating the inputs to *Move* without any obstacles or sloped terrain in front of the rover.

The dynamic weight adaptation seen in the ramp trials is shown in Figure 7, where the weights are shown for the *Go to Goal*, *Avoid Obstacles*, and *Preserve Motors* behaviors. There are rapid changes in the weights as the rover attempts to climb the ramp, followed by oscillations about a fixed point after numerous backing-off behaviors and then skirting the edge of the ramp. The variability in the weights over the trials is greatest when they are stabilizing to their new values (as seen in the size of the error bars). The eventual outcome of the sequence was that the rover learned to treat steeply sloped terrain as an obstacle, while at the same time trying to prevent motor burn-out. In the field, this behavior would be equivalent to the rover trying to find a safe way up a slope to get to the goal, as will be shown in the next sub-section.

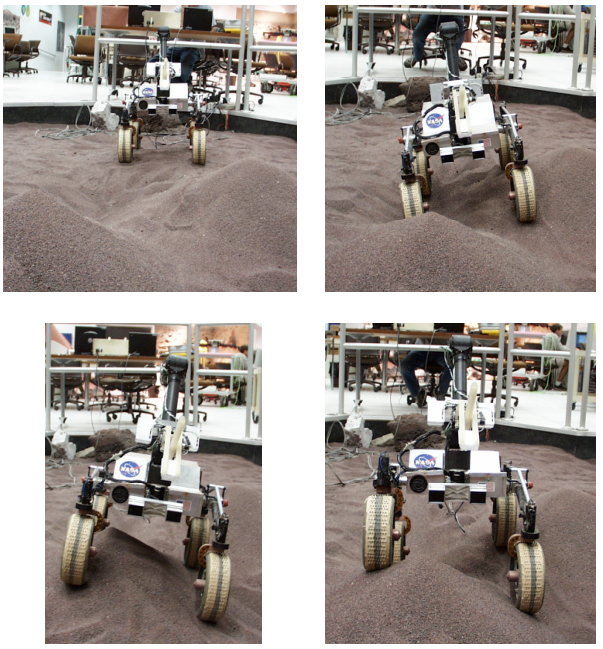

Figure 6. Clockwise from upper left: SRR performing continuous pose reconfiguration using its adjustable shoulders during a traverse in the Planetary Robotics Lab at JPL. The terrain was a set of two opposing hills and valleys, with 45º degree slopes.

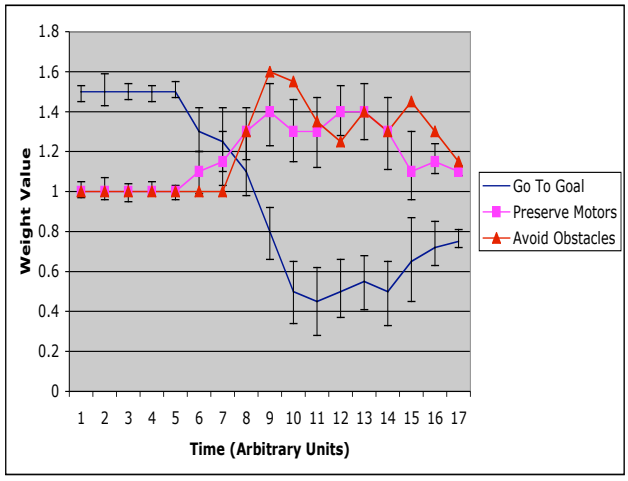

Figure 7. Adaptive learning of weights in the FFH for a rover attempting to go to a goal blocked by a steep ramp. Referring to Figure 2, the inputs to *Move* are *Go To Goal* (plain line), *Preserve Motors* (square line), and *Avoid Obstacles* (triangle line). Points on curves are the average over 20 trials with error bars giving the range of variation (error bars on *Avoid Obstacles* curve omitted for clarity).

C. *Field*

The last series of experimental studies was done in the Arroyo Seco, a dry wash that is next to JPL. This site is used for technology prototype rover testing and is characterized by a mixture of benign sand and rocky beds that have been scoured by the periodic water flow bounded by steeply sloped cliffs. The learning component of BISMARC was not fully implemented at the time, so only qualitative results are available at this time.

We were only able to complete a preliminary series of 4 runs in the Arroyo Seco and will return for more data collection in the end of September of 2004 prior to the winter rains. An example of the skirting behavior along a slope, as previously seen in the laboratory studies discussed in the previous sub-section, is shown in Figure 8, where the rover approaches the slope in the left frame and is not able to climb, skirts to the side in the middle frame, and finally gets enough traction to climb to the top of the rise and continue on towards the goal.

VI. Conclusions

We have provided a learning component to an autonomous rover control system for planetary rovers traversing rough and highly sloped terrain during long duration missions. The FFH action selection mechanism of BISMARC [5, 6] is coupled with adaptive onboard learning of weights in the hierarchy. The learning mechanism enabled BISMARC to maintain rover health in both simulated and actual rover studies in rough terrain. Of particular importance for future NASA rover missions was the analysis of the rover failures, indicating that an additional 52.9% of missions would potentially be successful with adaptive

Figure 9. Skirting behavior of SRR along the length of a slope in the Arroyo Seco wash outside of JPL where the rover initially can not get enough traction to climb so the direction of travel favors lateral motion. Time flows from left to right in the sequence with the left frame being the initial approach almost parallel to the slope, the middle frame showing a change in rover heading more perpendicular to the slope for better traction on the top of the rise, and the right frame showing the rover continuing along its initial heading towards the goal. The mast was fixed in its orientation for this run and would have given the rover more traction if pointed up slope.

onboard learning. We are currently optimizing the memory trace implementation and preparing for further trials in the Arroyo Seco (results should be collected in time for the meeting). We are also starting the integration of the BISMARC control techniques into the recently developed CAMPOUT (Control Architecture for Multi-robot Planetary Outposts) running on two technology prototype rovers at JPL [3, 45].

ACKNOWLEDGMENT

The authors would like to thank the members of the Planetary Robotics Laboratory at JPL for their support of this work. The opinions and conclusions are the authors', and are not meant to imply that they are those of JPL, the California Institute of Technology, or NASA. In addition, we would like to thank Steve Dubowsky and Karl Iagnemma from MIT for laying the groundwork for the reactive portion of the onboard control.

REFERENCES

[1] K. Iagnemma, A. Rzepniewski, S. Dubowsky, T. Huntsberger, and P. Schenker, "Mobile robot kinematic reconfigurability for rough-terrain," *Proc. Sensor Fusion and Decentralized Control in Robotic Systems III*, SPIE Vol. 4196, Boston, MA, 2000.

[2] K. Iagnemma, A. Rzepniewski, S. Dubowsky, and P. Schenker, "Control of robotic vehicles with actively articulated suspensions in rough terrain," *Autonomous Robots*, vol. 14, no. 1, pp. 5-16, 2003.

[3] P.S. Schenker, T.L. Huntsberger, P. Pirjanian, E.T. Baumgartner, and E. Tunstel, "Planetary Rover Developments Supporting Mars Exploration, Sample Return and Future Human-Robotic Colonization," *Autonomous Robots*, vol. 14, no. 2/3, pp. 103-126, 2003.

[4] P.S. Schenker, T. Huntsberger, P. Pirjanian, S. Dubowsky, K. Iagnemma, and V. Sujan, "Rovers for Intelligent, Agile Traverse of Challenging Terrain," *Proc. International Conference on Advanced Robotics (ICAR'03)*, University of Coimba, Portugal, pp. 1683-1692, 2003.

[5] T.L. Huntsberger and J. Rose, "BISMARC," *Neural Networks*, vol. 11, no. 7/8, pp. 1497-1510, 1998.

[6] T.L. Huntsberger, "Biologically inspired autonomous rover control," *Autonomous Robots,* vol. 11, no. 11, pp. 341-346, 2001.

[7] T.L. Huntsberger and H. Aghazarian, "Learning to Behave: Adaptive Behavior for Planetary Surface Rovers," *Proc. 8th International Conf. on Simulation of Adaptive Behavior (SAB04), From Animals to Animats 8,* Los Angeles, CA, July, 2004.

[8] T.L. Huntsberger, "Autonomous multi-rover system for complex planetary surface retrieval operations," *Proc. Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, SPIE Vol. 3209, pp. 220-229, 1997.

[9] T.L. Huntsberger, **"**Fault-tolerant action selection for planetary rover control," *Proc. Sensor Fusion and Decentralized Control in Robotic Systems*, SPIE Vol. 3523, pp. 150-156, 1998.

[10] T.L. Huntsberger, M.J. Mataric, and P. Pirjanian, **"**Action selection within the context of a robotic colony," *Proc. Sensor Fusion and Decentralized Control in Robotic Systems II*, SPIE Vol. 3839, pp. 84-91, 1999.

[11] E. Tunstel, "Ethology as an inspiration for adaptive behavior synthesis in autonomous planetary rovers," *Autonomous Robots,* vol. 11, no. 11, pp. 333-340, 2001.

[12] N. Tinbergen, *The Study of Instinct*, Oxford University Press, 1951.

[13] G.P. Baerends, "A Model of the Functional Organization of Incubation Behaviour in the Herring Gull," *Behaviour, An International Journal of Comparative Ethology*, Suppl. 17. Leiden-E. J. Brill, pp. 261-310, 1970.

[14] P.D. MacLean, *A Triune Concept of the Brain and Behavior*, University of Toronto Press, Toronto, Ontario, 1973.

[15] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14-23, 1986.

[16] T. Tyrrell, "The use of hierarchies for action selection," *Journal of Adaptive Behavior*, vol. 1, no. 4, 1993.

[17] J.J. Bryson, "Hierarchy and Sequence vs. Full Parallelism in Action Selection," *From Animals to Animats 6, Proc. of the Sixth Intern. Conf. on Simulation of Adaptive Behavior (SAB'00),* pp. 147-156, 2000.

[18] J.K. Rosenblatt and D.W. Payton, **"**A fine-grained alternative to the subsumption robot control," *Proc. IEEE/INNS Joint Conf. on Neural Networks*, pp. 317-324, 1989.

[19] J.J. Bryson, *Intelligence by Design*. Ph.D. dissertation, Dept. of Elec. Engineering and Computer Science, MIT, Cambridge, MA, USA, 2001.

[20] P. Maes, (Ed.) *Designing Autonomous Agents Theory and Practice from Biology to Engineering and Back*, MIT Press, 1991.

[21] L.P. Kaelbling, *Learning in Embedded Systems*. MIT Press, Cambridge, MA, USA, 1993.

[22] L.P. Kaelbling, M.L. Littman, and A.W. Moore, "Reinforcement Learning: A survey," *J. of Artificial Intelligence Research*, vol. 4, pp. 237-285, 1996.

[23] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[24] C.J. Watkins, *Learning from Delayed Rewards*. Ph.D. dissertation, Cambridge Univ., Cambridge, UK, 1989.

[25] T. Horiuchi, A. Fujino, A., Katai, O., and T. Sawaragi, "Q-PSP Learning: An exploitation-oriented Q-learning algorithm and its applications," *Proc. IEEE International Sympos. on Evolutionary Computation*, pp. 76-81, 1996.

[26] L.-J. Lin, *Reinforcement Learning for Robots Using Neural Networks*. Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA, 1993.

[27] C. Gaskett, D. Wettergreen, and A. Zelinsky, "Q-learning in continuous state and action spaces," *Proc. 12th Australian Joint Conf. on AI*, Sydney, Australia, 1999.

[28] Y. Takahashi, M. Takada, and M. Asada, "Continuous valued Q-learning for vision-guided behavior acquisition," In *Proc. International Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp. 716-721, 1999.

[29] M. Takeda, T. Nakamura, M. Imai, T. Ogasawara, and M. Asada, M. (2000). "Enhanced continuous valued Q-learning for real autonomous robots," *From Animals to Animats 6, Proc. of the Sixth Intern. Conf. on Simulation of Adaptive Behavior (SAB'00)*, 2000.

[30] G. Yen, F. Yang, T. Hickey, and M. Goldstein, "Coordination of exploration and exploitation in a dynamic environment," P*roc.*

International Joint Conference on Neural Networks (IJCNN '01), Vol. 2, pp. 1014-1018, 2001.

[31] G. Yen and T. Hickey, "Reinforcement learning algorithms for robotic navigation in dynamic environments," In *Proc. of the 2002 Intern. Joint Conf. on Neural Networks (IJCNN '02), Vol. 2*, pp. 1444-1449, 2002.

[32] I.H. Suh, M.J. Kim, S. Lee, and B.J. Yi, "Design and Implementation of a Behavior-Based Control and Learning Architecture for Mobile Robots" *Proc. 2003 IEEE International Conf. on Robotics and Automation (ICRA2003)*, Taiwan, pp. 4142-4147, 2003.

[33] I.H. Suh, M.J. Kim, S. Lee, and B.J. Yi, "A Novel Dynamic Priority-Based Action-Selection-Mechanism Integrating a Reinforcement Learning," *Proc. 2004 IEEE International Conf. on Robotics and Automation (ICRA2004)*, New Orleans, LA, pp. 2639-2646, 2004.

[34] M. Humphrys, *Action Selection Methods using Reinforcement Learning*. PhD thesis, University of Cambridge, Cambridge, UK, 1997.

[35] B.D. Hoffman, E.T. Baumgartner, T. Huntsberger, and P.S. Schenker, "Improved rover state estimation in challenging terrain," *Autonomous Robots,* vol. 6, no. 2, pp. 113-130, 1999.

[36] M.J. Mataric, "Integration of representation into goal-driven behavior-based robots," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 304-312, 1992.

[37] M.J. Mataric, "Behavior-based control: Examples from navigation, learning, and group behavior," *Journal of Experimental and Theoretical Artificial Intelligence, Special Issue on Software Architectures for Physical Agents*, vol. 9, no. 2-3, pp. 323-336, 1997.

[38] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 3, pp. 249-265, 1987.

[39] S. Mahadevan and J. Connell, "Automatic Programming of Behavior Based Robots Using Reinforcement Learning," *Artificial Intelligence*, vol. 55, pp. 311-365, 1992.

[40] M.J. Mataric, "Reinforcement Learning in the Multi-Robot Domain," *Autonomous Robots,* vol. , no. 1, pp. 73-83, 1997.

[41] A.K. McCallum, *Reinforcement Learning with Selective Perception and Hidden State*. PhD dissertation, Department of Computer Science, Univ. of Rochester, 1995.

[42] A.K. McCallum, "Learning to Use Selective Attention and Short-Term Memory in Sequential Tasks," *From Animals to Animats 4, Proc. Fourth International Conf. on Simulation of Adaptive Behavior, (SAB'96)*, Cape Cod, MA, 1996.

[43] F. Michaud and M.J. Mataric, "Representation of behavioral history for learning in nonstationary conditions," *Robotics and Autonomous Systems,* vol. 29, pp. 187-200, 1999.

[44] F. Michaud and M.J. Mataric, "Learning from History for Behavior-Based Mobile Robots in Non-stationary Conditions," *Joint Special Issue on Learning in Autonomous Robots, Machine Learning,* vol. 31, no. 1-3, pp. 141-167, and *Autonomous Robots,* vol. 5, no. 3-4, pp. 335-354, 1998.

[45] T. Huntsberger, P. Pirjanian, A. Trebi-Ollennu, H.D. Nayar, H. Aghazarian, A. Ganino, M. Garrett, S.S. Joshi, and P.S. Schenker, "CAMPOUT: A Control Architecture for Tightly Coupled Coordination of Multi-Robot Systems for Planetary Surface Exploration," *IEEE Trans. Systems, Man & Cybernetics, Part A: Systems and Humans, Special Issue on Collective Intelligence,* vol. 33, no. 5, pp. 550-559, 2003.